Goma 6.1 - A Full-Newton Finite Element Program for Free and Moving Boundary Problems with Coupled Fluid/Solid Momentum, Energy, Mass, and Chemical Species Transport: Users Guide

P. Randall Schunk, Rekha R. Rao, Ken S. Chen, Duane A. Labreche, Amy C. Sun, Matthew M. Hopkins, Harry K. Moffat, R. Allen Roach, Polly L. Hopkins, Patrick K. Notz, Philip A. Sackinger, Samuel R. Subia, David R. Noble, and Scott A. Roberts
Sandia National Laboratories
Albuquerque, NM 87185

Thomas A. Baer
Gillette
Boston, MA 02127

Edward D. Wilkes
Corning Inc.
Corning, NY 14830

Robert B. Secor
3M Company
St. Paul, MN 55144

Kristianto Tjiptowidjojo, Andrew M. Cochrane, and Weston Ortiz
University of New Mexico
Albuquerque, NM 87131

Abstract

Goma 6.1 is a finite element program which excels in analyses of multiphysical processes, particularly those involving the major branches of mechanics (viz. fluid/solid mechanics, energy transport and chemical species transport). *Goma* is based on a full-Newton-coupled algorithm which allows for simultaneous solution of the governing principles, making the code ideally suited for problems involving closely coupled bulk mechanics and interfacial phenomena. Example applications include, but are not limited to, coating and polymer processing flows, super-alloy processing, welding/soldering, electrochemical processes, and solid-network or solution film drying. This document serves as a users guide and reference.

Preface

Over the course of development of this new generation of *Goma* documentation, the volume of information collected between the covers has grown immensely while the style of presentation of that information has also been improved to be more helpful to the analyst and easier to use. However, having set the goal of producing both a printed and electronic manual, the process has made it no longer practical to try to contain all the attending knowledge in a single printed volume. Thus, we have divided the printed version along the boundaries most natural to *Goma*, that being a separation according to the division of problem data between the two primary ASCII input files.

The user of *Goma* software now has a two-volume manual with information both common and unique to each volume. The introductory information (Chapters 1 through 3) is common to both volumes, as is the closing information (References, Appendix and Distribution). The unique contents of Volume 1 consist of the Problem Definition (Chapter 4), while Volume 2 contains the Material File description (Chapter 5). In the respective locations of the Chapter 4 and 5 information, a brief explanatory note has been inserted as a placeholder. The user will find a complete set of introductory and closing information in each volume, but the Table of Contents and Index entries in each volume will also be unique, containing only the information appropriate for the particular volume.

Also for practical reasons, this electronic version of the manual will retain the single volume configuration. Thus the structure will differ from the printed version but the contents of the two versions of the manual will contain the same information.

Acknowledgement

Development of *Goma* was funded in part by the Engineering Science Research Foundation, Laboratory Directed Research and Development, the Coating and Related Manufacturing Processes Consortium (CRMPC), the Specialty Metals Processing Consortium (SMPC), the Accelerated Scientific Computing Initiative (ASCI) program of the DOE, and the Basic Energy Science Program of the DOE. The authors would like to thank Rick Givler, Mike Kanouff, Anne Grillet, Mark Christon, John Torczynski, and many others for their helpful comments during the process of developing the code and reviewing this manual. The third and fourth editions of this manual have benefited from several others. Namely, Chris Monroe, who took over the responsibility for updating and distributing the second edition of the report, and several internal Sandia users together with several members of the Coating and Related Manufacturing Processes Consortium who provided valuable feedback.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Goma, which means *rubber*, *gum*, or *elastic* in Spanish, is a two- or three-dimensional finite element program currently being advanced and specialized for the analysis of manufacturing flows and related processes that involve one or more transport fields, i.e., any combination of heat, mass, momentum (solid and fluid) and species transport fields. Specifically, the processes for which *Goma* is suited are those which contain free or moving boundaries between dissimilar materials or phases. Whether determining the position of an interface whose motion is governed by the underlying physics of the problem, or prescribing the dynamics of a boundary according to user specified kinematics or geometry, the multiphysics approach on which *Goma* is based allows for rapid convergence to the solution. Unique features which make this possible include: (1) a Lagrangian-Eulerian solid mechanics module for mesh motion, (2) energy and chemical species transport modules incorporating convection, diffusion and reaction, (3) fluid momentum transport modules that are fully and mutually coupled, particularly with the mesh motion module through an analytical Jacobian matrix, (3) a Newton-based solution algorithm (full and modified) which exploits that Jacobian matrix, and (4) a structure which allows for different physical descriptions of different materials in the same problem, i.e., conjugate problems. The scope of potentially accessible problems defined by the interaction and close coupling of the individual field equation sets is partially shown in Figure 1 (note that missing from this figure are the fully coupled, partially saturated porous deformable media module and overall variable density mass balance modules). The analytical Jacobian matrix which provides the coupling facilitates a range of computer-aided nonlinear analyses such as parametric sensitivity (stability), design, and optimization as it provides the building blocks (through chain-rule differentiation) for evaluating sensitivities of process variables to processing conditions.

Goma originated in 1994 from an early version of MP SALSA (Shadid et al., 1995), a finite element program designed to simulate chemically reacting flows in massively-parallel computing environments. As a point-of-departure, Goma was originally extended and adapted to free and moving boundary problems in fluid mechanics, heat transfer, and mass transfer. By virtue of a novel mesh motion algorithm based on Lagrangian solid elasticity, many multiphysics problems involving nonlinear elasticity and viscoplasticity in combination with other transport phenomena are now accessible. The detailed algorithm and underlying physical principles of the moving mesh scheme together with several advanced examples from capillary hydrodynamics, melting and solidification, and polymer processing may be found elsewhere (Schunk and Rao, 1994; Cairncross et al., 1995; Chen et al., 1995; Sackinger et al., 1996; Bertram et al., 1998; Cairncross et al., 2000; Baer et al., 2000; Schunk et al., 2001)

Since the original publication of the GOMA 2.0 manual (see Schunk et al. 1997), work has further focused on concentrated chemical species transport (neutral and charged species) and Eulerian front tracking schemes for large material deformation problems. As in all other developments, these capabilities are being implemented in a fully-coupled way using Newtons method. A concerted effort to bring these capabilities to bear on real-life problems has led to the addition of many esoteric features that address capillary wetting, phase change, charge neutrality, multicomponent species transport, and a host of other physical features. The best way to survey the available features is to consult the large library of reports, technical memoranda, tutorials, and other advanced feature manuals (e.g. Schunk 2000; Gates et al. 2000; Rao et al. 2001; see Goma Documentation List in the Appendix), most of which are linked together with this manual in the CD version of the Goma Document System currently under development.

Most recent developments, from 2006 through 2012, that are noteworthy are an extensive library of thin-shell physics/equations and accompanying boundary conditions, triangle and tetrahedral elements, phase-field modeling, parallel processing improvements and more. On the thin shell equations, the capability is fully coupled with continuum element equations. We have implemented theory and equations for Reynolds lubrication, thin-shell energy, thin-porous media, and surface rheology.

The purpose of this report is to provide a practical introduction and reference to Goma; to introduce the user to the range of options available in Goma; to show how easily the code may be adapted to investigate novel situations; and to provide a link to several simple illustrative examples as a tutorial and as a demonstration of the overall utility of the program. By design this is a reference manual which is best navigated together with a tutorial on the class of problems being addressed. It is recommended that perusal be undertaken section by section, consulting the individual input records as needed for a given problem.

Chapter 2

Background Information

2.1 Program Features

2.1.1 Free and moving boundary capabilities

Goma is a general purpose program designed for the solution of both steady and transient, two and three-dimensional problems involving heat, mass, and momentum (solid and fluid) transport. A unique feature is the treatment of all boundaries and interfaces as *free* (position unknown) or moving (position unknown or prescribed, but variable). If the material domain of interest is a solid, a Lagrangian formulation (i.e., the computational mesh follows the motion of material) of the momentum equations naturally leads to mass conservation and a natural parameterization of the boundaries and interfaces as material surfaces. If the material domain of interest is a fluid, then an Arbitrary-Lagrangian-Eulerian (ALE) formulation allows the boundaries to respond to constraint equations, hereafter referred to as *distinguishing conditions*. These conditions are responsible for determining the location of all boundaries and interfaces, providing the necessary mathematical closure of the system of equations governing the free boundary problem. Distinguishing conditions available to the user fall into several classes, as described below.

Since publication of the *Goma* 2.0 manual in 1998 (and more recently the Goma 4.0 manual in 2002), the ALE formulation has been extended to solid-material regions (viz. the TALE algorithm, Schunk 2000) and purely Eulerian front tracking schemes based on the method of level-sets have been incorporated for free surfaces with large deformations; moreover, both algorithms have been implemented in a completely-coupled way. Of course Eulerian schemes are inherently transient and less accurate in capturing interfacial physics, even though they are more robust and even optimal for a certain class of problems. It is fair to say that of all the available mechanics codes, Goma provides the greatest breadth of free and moving boundary tracking formulations and options.

With regard to the ALE algorithms, the fully-implicit, pseudo-solid, unstructured mesh deformation algorithm sets *Goma* apart from other finite element programs. All surfaces, internal and external, together with other geometric features such as corners and junction points, are permitted to move as part of the algorithm. The movement of boundaries, interfaces, and geometric features is dictated by a weighted residual statement of the distinguishing conditions, whether based on specific physical constraints or arbitrary conditions described by the analyst. The internal mesh deforms as if it were embedded in a deforming elastic solid continuum; with the mechanics of the solid governed by either infinitesimal (linear) or finite (nonlinear) deformation theory. Through Newtons method, the deformation of the mesh is determined simultaneously with all of the other physics of the problem.

The key connection between the mesh deformation and the physics of interest is accomplished through a library of distinguishing conditions. Currently, those conditions include (a) kinematic (material surface of a fluid), (b) isotherm (phase transition temperature, such as melting), (c) isoconcentration and (d) geometric (either smooth plane curves or fixed point specifications). As part of the required input for *Goma*, the analyst specifies the associations between the particular distinguishing conditions and corresponding sets of material points of the initial pseudo-solid used to embody the mesh. Chapter 4 describes this process in more detail. Essentially, the algorithm causes smooth boundaries of the pseudo-solid to slide tangentially in a frictionless fashion. Further details of this algorithm and the corresponding equations can be found in several references (e.g., Sackinger et al. 1996).

2.1.2 Coordinate Systems and Frames of Reference

Coordinate systems accessible through this version of *Goma* include two-dimensional and three-dimensional Cartesian coordinates, cylindrical coordinates for axisymmetric problems, spherical coordinates, and a swirling option for two-dimensional axisymmetric problems with a (swirling) velocity component in the third dimension. A limited framework has been built within *Goma* to use arbitrary orthogonal curvilinear coordinate systems, but this has not yet been extensively tested. As for frame of reference, all conservation equations are cast in an inertial frame (viz. nonaccelerating) but with extensions to allow for arbitrary frame velocities that may or may not be related to the material motion. Hereafter, when we refer to the frame/mesh motion type to be of the Eulerian variety, we mean the mesh is fixed with respect to all material motion, which basically means it is fixed in the laboratory frame. For now, we allow this frame of reference for fluid systems and are researching ways to allow this frame for solid systems. The ALE frame of reference, as mentioned above, allows for independent mesh motion in the interior of the domain, but seeks to maintain a material frame of reference on the boundary. This means that the mesh will move to accommodate material boundary motion. Currently, the ALE frame is allowed for all classes of materials (cf. Schunk 2000). Finally, a pure Lagrangian frame of reference implies that our mesh moves with the material. This formulation is quite common in solid mechanics and is one advocated here for truly solid regions.

2.1.3 Problem Physics and Thermophysical Properties

This brief section summarizes the physics capabilities in *Goma* and the thermophysical properties and constitutive equations available to the user. The rest of the manual is designed to greatly expand on all material parameter options, boundary condition options, and equation options; perusing Chapter 4 and Chapter 5 is recommended to extract more detail.

The class of problems treated by *Goma* are those described by any one or a combination of the incompressible form of the momentum conservation equation for generalized Newtonian fluids, the momentum conservation and differential stress constitutive equations for viscoelastic fluids, saturated and unsaturated flow equations cast for rigid or deformable porous media, the energy conservation equation, the equations of quasi-static equilibrium of an elastic solid, and any number of additional or auxiliary species convection-diffusion-reaction equations. *Goma* has been tested with the following types of fluid mechanics, solid mechanics, and heat transfer problems: (a) mixed convection with mesh parameterization of an isotherm, (b) melting, with a parameterization of the liquidus and solidus isotherms, (c) coating and related flows (slide coating, curtain coating, etc.), (d) polymer processing (viscoelastic) flows (e.g. fountain flow, planar and axisymmetric extrusion, simple mold filling, contraction flow), (e) neutral or charged species transport in multicomponent concentrated systems, (f) partially saturated flow in poroelastic systems, (g) suspension flows, (h) drying and shrinking of gelled polymer films (with creep and elastic recovery), and (i) microfluidic systems with fluid-structure interaction (e.g. MEMS device performance).

Thermophysical properties in the bulk for all equations may be taken as constant or variable, with dependencies on any of the dependent and independent variables of the problem. General property variation models of this sort can be implemented with a user-defined subroutine capability. Moreover, a growing number of often-used standard models are supported within the core routines. These include a Carreau-Yasuda model for the generalized Newtonian viscosity and a Boussinesq source term for the fluid momentum equation that provides a means for simulating flows with thermal and solutal buoyancy. A plethora of other constitutive models and properties are available, including viscoelasticity, elastoviscoplasticity, nonFickian diffusivity, etc.

To enhance the capability for modeling problems in capillary hydrodynamics, e.g., coating flows, a boundary condition expressing the normal stress balance for two-dimensional Cartesian and axisymmetric problems has been implemented and tested. When capillary forces are activated, a pressure jump term (proportional to the mean curvature) is added to the normal component of the momentum flux balance at specified fluid material interfaces in a natural fashion. At three-phase boundaries (points in two dimensions) a contact angle condition and a surface tangent force condition may be applied. The former is used in place of a specified position on the mesh motion equations and is best used to set static and dynamic contact angles, and the latter is an additional endpoint force which is added to the momentum balance, necessitated because the curvature term is integrated by parts. The current version of *Goma* also includes the ability to model tangential shear forces along capillary surfaces, i.e., those originating from surface tension gradients caused, for example, by variations in temperature or species concentration. To access this capability requires a constitutive equation for the surface tension. A powerful low-level capability has been implemented which allows the user to select which degree of freedom, or variable, is associated with a particular boundary condition. Such a capability is useful at dynamic contact lines, where it is often desirable to replace the liquid-phase momentum equations with auxiliary constraint conditions.

Generalized interphase boundary conditions that allow for discontinuous field variables are supported through a multiple degree-of-freedom capability. The prime targets for this capability include flowing vapor-liquid equilibrium problems for which there are concentration and velocity jumps between phases due to change in density and solute partitioning through the phase diagram and multiphase/multicomponent corrosion problems. A series of boundary conditions which allow for the application of ideal and non-ideal vapor/liquid equilibrium (e.g. Raoults law and Flory-Huggins theory), latent heat release/adsorption, and discontinuous velocity components due to evaporation/condensation have been implemented. In the future this capability can be extended to thermal contact resistance, which often involves a temperature jump at an interface.

Recently the solid mechanics module of *Goma*, which was originally installed as a part of the pseudo-solid ALE mesh motion algorithm, has been exploited to solve problems in transport in deformable porous media and other outstanding problems of elastohydrodynamics. For modeling flow in non-deformable porous media, the Brinkman terms in the fluid momentum equations (cf. Gartling et al. 1996) may be activated. Since *Goma* 2.0, generalized Darcy transport equations for multiphase components (solid, liquid, gas) have been added and can be used for simulations of deformable poroelastic media. For incompressible but deformable solids, a pressure term was added to the solid momentum balance (e.g. rubber). In continuous shrinking or swelling solids, the dilation is proportional to changes in solvent concentration. In deformable porous media, the solid deformation is coupled to the pressure in the fluid-filled interstices of the porous matrix. Several boundary conditions exist to apply normal tractions (i.e. compressive, tensile, or shear boundary forces) to solid surfaces. To effectively simulate coupled fluid/solid interaction problems, boundary conditions which balance the surface tractions exerted by the liquid and solid phases at the common interface have been incorporated as have been the appropriate interface impregnation/expulsion conditions at boundaries between porous and continuous media.

A complete rewrite of the species transport equations has been undertaken since the release of *Goma* 2.0 that allows for generalized phase/species formulations on multimaterial problems. Accommodating an arbitrary number of species, each of which can exist in an arbitrary number of phases, was the goal of this development in order to model corrosion and charged species transport.

Of course there are many more material property models and constitutive equations, specialized boundary conditions, and more esoteric differential equations that can be solved for just about any mechanics problem. Many of these capabilities are not cited in this manual because they were under development at the time of publication. Interested readers should inquire about the status of the following capabilities: generalized solid-model geometry features, wetting and spreading models for Eulerian front tracking schemes, Eulerian/Eulerian fluid-structural interaction capability, multiphase porous energy equation, Generalized surface and volume user-defined Lagrange multiplier constraints, and much more.

2.1.4 Advanced capabilities

Several developments in Goma that enable advanced engineering analysis of complex systems have been completed since the last major release. These developments include a complete, generalized capability of automated parameter continuation (zeroth-order, first-order, arclength, multiparameter, user-defined parameter continuation, etc.) using the LOCA library (Salinger et al., 2002), linear stability analysis of any dynamic system using normal modes, and augmenting condition capability. It is recommended that the user consult a separate accompanying manual (Gates et al. 2000 CITE NEWEST ADVANCED CAPABILITIES MANUAL) for a complete user description of these features. The input record sections required to activate these features are not covered in this document.

2.2 Numerical Methods

With over 200 different boundary conditions for 180 plus differential equation types UPDATE THESE NUMBERS, *Goma*s algorithms are very extensive for any brief discussion. In this section we simply point out the foundation algorithms. A developers manual, advanced capabilities manual, and tutorial memos can be consulted for more details (see Goma Document List in the Appendix for the citations.).

Goma is based primarily on the Galerkin/finite element method. The element library currently includes (in two dimensions) 4- and 9-node isoparametric quadrilaterals (i.e., Q1 and Q2 interpolations) with available interpolations for linear discontinuous (P1) or piecewise constant (P0) variables, and (in three dimensions) 8-node isoparametric hexahedral elements and 27-node bricks, also available with piecewise constant interpolations. INCLUDE TRIANGLES AND TETRAHEDRALS AS WELL. The overall solution algorithm centers around a fully-coupled Newton-Raphson iterative scheme for solving the nonlinear algebraic equations which results from the finite element discretization. That is, all active equations and boundary conditions are solved simultaneously in a single matrix system at the same time plane and during the same Newton iteration. The sparse matrix system is stored in a central element-level matrix data structure that is injected into one of three sparse matrix formats as dictated by the matrix solver chosen. The three formats are modified sparse row, MSR or compressed row format (Hutchinson et al., 1995; Schunk and Shadid, 1992), the variable block row, or VBR, format (see Heroux 1992), or the frontal-solver element-level format (cf. Hood 1976). ADD EPETRA MATRIX FORMAT HERE TOO. If the matrix system is not too poorly conditioned, then iterative solvers of the generalized preconditioned conjugate gradient-type can be used to solve the system (see Schunk and Shadid 1992; Tuminaro et al. 1999). A matrix-services/solver-services library known as Trilinos (`https://trilinos.org`), has been installed to handle all iterative solver and preconditioner options. This package has greatly extended the robustness of iterative solvers to the class of problems that *Goma* solves. Virtually all problems and all finite element formulations are now solvable with these iterative schemes (see Schunk et al. 2001). If all else fails, Goma deploys a suite of direct solvers that, even though not always efficient for large three-dimensional problems, will always get a solution at the current Newton iteration. These solvers are Sparse 1.4 (`lu`), a classical LU decomposition (Gaussian elimination) method and frontal solver UMFPack (`umf`). For direct solvers utilizing parallel processing, *Goma* is able to employ SuperLU (`superlu`) and MUMPS (`mumps`) via Trilinos package.

The Galerkin least squares (GLS) method for pressure stabilization of Hughes and Franca (1987) has also been added to Goma. The GLS method adds the momentum residual, weighted by the gradient of the Galerkin weight function, to the standard Galerkin continuity equation, thus providing a diagonal term for the pressure. This is a first-order convergent and consistent method that enables the use of iterative solvers for incompressible equations over the entire range of Reynolds numbers. NEED TO ADD SUPG AND GLS FOR VISCOELASTIC FLOW. SHOULD WE ADD DG AS WELL?

The overall differential-algebraic system of equations may be advanced in time with implicit time-integration techniques (simple backward Euler and Adams-Bashforth predictor, trapezoidal corrector algorithms for fluid systems, species transport and energy transport; and Newmark-Beta algorithms for solid dynamics). Time marching offers an alternative, albeit indirect, route to attaining solutions to steady equations, as well as providing the capability of simulating process transients directly. Automatic time step control based on current truncation error is also available. SEGREGATION MATRIX HERE?

Perhaps the most complicated part of the algorithm is the construction of the Jacobian sensitivity matrix. Because the mesh point positions are actually unknowns in a free or moving boundary problem, that matrix must include sensitivities of each weighted residual equation with respect to each of the mesh variable unknowns that can affect the value of the residual. Unfortunately, almost every term of the bulk equations and many boundary conditions contribute to this sensitivity. This occurs mainly through gradient operators and surface normal and tangent vectors (see Kistler and Scriven 1983) and through dependencies on mesh position of the determinant of the elemental Jacobian transformation matrix that maps between a fixed unit element and any element in the computational domain. Great care has been taken to include analytical expressions for all of these mesh sensitivities. However, some of this task inevitably falls to the user when implementing user-defined boundary conditions, material property models, and constitutive equations, particularly when any of these quantities depends directly on spatial position or spatial gradients of other variables. In order to maintain the strong convergence properties of Newtons method, these sensitivities must be specified in those user-defined routines. To aid in this task, a debugging option is available which computes a numerical finite-difference approximation of the global Jacobian matrix and compares it with its analytical counterpart. This tool enables users and developers to check the consistency of newly-created equations (whether bulk or boundary constraints) with their corresponding analytic Jacobian contributions.

2.3 Portability, Software Library Infrastructure, and Code Accessibility

Goma is written in the C programming language (specifically Kernighan and Ritchie (1988) C with some ANSI extensions). It has been ported to a number of UNIX platforms including Solaris and Linux, with the Red Hat Linux Enterprise-6 and 7 versions being the most actively maintained. Many of the machine dependencies in the program have been isolated using C preprocessor directives. Some of the machine dependencies that occur in the I/O routines are insulated from the user by software libraries. Building Goma requires EXODUS II (Schoof and Yarberry, 1992), SPARSE 1.4 (Kundert, 2003), NetCDF (`http://www.unidata.ucar.edu/software/netcdf/`) libraries, UMFPack direct solver libraries (Davis and Duff, 1997), and the TRILINOS library (`https://trilinos.org`). The first of these is part of the SEACAS system at Sandia National Laboratories (`http://gsjaardema.github.io/seacas/`). All of the packages are available publicly. Parallel processing is enabled by OPEN-MPI. The user should consult the build instructions for the most recent library revisitions. The most updated library needs are also made clear in the Goma makefile: `Makefile`. TO DEBUG USE MAKE DEBUG. Generally, pre- and post-processing is performed outside of *Goma*, although some post-processing of results is available within the program. This separation of the functionality permits the use of alternative solid-modeling and mesh-generation software and visualization packages of choice, insofar as they may be interfaced with the EXODUS II finite element data model. NEED TO UPDATE TO THE LATEST BUILD PROECEDURE.

Pre-processing options include mesh generation most commonly created via CUBIT (`https://cubit.sandia.gov`), or Trelis (`http://www.csimsoft.com/trelis`). PATRAN (`http://www.mscsoftware.com/product/patran`), and SolidWorks (`www.solidworks.com`) can also be used but require special plug-ins. These mesh generators need to support and output a finite element database in the EXODUS II format. UPDATE TO USE OF MOSES FOR GMSH.

Post-processing options include BLOT, which is part of SEACAS distribution (`http:`

`//gsjaardema.github.io/seacas/`), Paraview (`http://www.paraview.org`), and Ensight (`https://www.ensight.com`).

Since *Goma* is built around the EXODUS II finite element data model, there are numerous options available for communication with other analysis codes that also exchange data via the same EXODUS II data model. Recent modifications to *Goma* permit not only the initialization of unknown values from an EXODUS II file, but also the ability to incorporate field variables into the analysis that are not unknowns. For example, the quasi-static and dynamic electromagnetic fields from codes such as ALEGRA can be used to compute electric fields and current fluxes on a specified finite element mesh that are input to *Goma* through the EXTERNAL FIELD data card.

Chapter 3

Code Structure and I/O

3.1 Files for Data Input

The *Goma* file I/O structure is diagrammed in Figure 2. Input to the program is divided into six categories: (1) command-line options, (2) problem description file, (3) material files, (4) ASCII continuation/restart file, (5) EXODUS II database file, and (6) sundry material property or boundary condition table lookup files. *Goma* is basically set up to run in batch mode, i.e., no input is required on the command line or after the run command is issued. There are, however, several command-line switches which can be used to redirect I/O, control the level of I/O, and activate debugging options.

The *problem-description file* is by default called input but can be renamed with the **-i** switch on the command line. A version of this file is also output as an echo file, viz. a prefix echo prepended to the input file name. The echo file is used to verify input into goma, as it clearly states all default settings for the input file and material files. . The input file itself contains the general description of the problem and directions to Goma on how to solve it (see Chapter 4). The file is split into fifteen sections: (1) File Specifications (Section 4.1) which directs I/O, (2) General Specifications (Section 4.2), (3) Time Integration Specifications (Section 4.3), (4) Continuation Specifications

(Section 4.4), (5) Hunting Specifications (Section 4.5), (6) Augmenting Condition Specification (Section 4.6), (7) Solver Specifications (Section 4.7), (8) Eigensolver Specifications (Section 4.8), (9) Geometry Specification (Section 4.9), (10) Boundary Condition Specifications (Section 4.10), (11) Rotation Specifications (Section 4.11), (12) Problem Description (Section 4.12), and (13) Post Processing Specifications (Section 4.13); this latter section includes breakouts for fluxes and data (Section 4.14), particle traces (Section 4.15) and for volume-based integrals. The file format is described in detail in Chapter 4. Incidentally, the structure of the data input routines is divided roughly along the same lines as the input data file itself.

The *material description* files (using the nomenclature [material name].mat) contain all material property data and material property model and constitutive model specifications. The names of these files are specified in the problem description file. The format of these files and the available options are described in Chapter 5. Note that these files are also reproduced as output as echo files, with all default settings specified.

The *ASCII continuation/restart files* (may have any name) contain an ASCII list of the solution vector (values of field variables at nodes), which can be used as an initial guess for successive runs of *Goma*. The names of these files are specified in the problem description file, but may be changed with the **-c** (for input) or **-s** (for output) command-line options. These restart files are recyclable, in the sense that output from one *Goma* simulation may be used as input to another *Goma* simulation under certain restrictions.

The *EXODUS II database files* (may have any name but generally end in .exoII) contain a description of the finite-element structure for the current problem. All EXODUS II files contain a definition of the mesh, material blocks, and boundary sets. In the case of input EXODUS II files created from mesh generator output, this is the sole content of the file. Output EXODUS II database files contain a clone of the input EXODUS II mesh information and also contains the nodal values of all field variables in the solution. The names of these files are specified in the problem description file, but may be changed with the **-ix** (for input) or **-ox** (for output) command-line options. The only EXODUS II file required when running Goma is the one containing the current problem mesh. All others are either output for postprocessing or used to supply auxiliary external fields (e.g. magnetic fields).

Chapter 4

Data Input – Problem Description File

4.1 File Specifications

In general, this first section of the main input file is used to direct *Goma* I/O through a series of named external files that contain information about the finite element mesh, the initial guess of a solution vector, and output options for saving solutions for continuation, remesh, etc. The required and optional input records are as follows:

4.1.1 FEM file

```
FEM file = <file_name >
```

Description/Usage This required card specifies the name of the EXODUS II finite element mesh file. Any EXODUS II file name is permissible, as specified below.

<file_name >	A file name of the form prefix.exoII. The prefix portion is any user-specified alpha-numeric string, which can be used as a problem-type descriptor. Preprocessors and postprocessors (like AVS) might require the .exoII suffix so it is a required part of the file designation. The maximum length of the file name is 85 characters

Examples Following is a sample card: `FEM file = in.exoII`

Technical Discussion

Theory

FAQ

References

Bibliography

Baer, T. A., Schunk, P. R., Cairncross, R. A., Rao, R. R. and Sackinger, P. A. 2000 A finite element method for free surface flows of incompressible fluids in three dimensions. Part II. Dynamic wetting lines. *Int. J. Numer. Meth. Fluids* **33** 405–427.

Bertram, L. A., Schunk, P. R., Kempka, S. N., Spadafora, F. and Minisandram, R. 1998 The macroscale simulation of remelting processes. *J. of Minerals, Metals, and Materials Society* **50** 18–21.

Cairncross, R. A., Chen, K. S., Schunk, P. R., Brinker, C. J. and Hurd, A. J. 1995 Recent advances in theoretical modeling of deposition, drying, and shrinkage in sol-gel coating processes. Proceedings on Computational Modeling of Materials and Processing Symposium at the American Ceramic Society National Meeting Cincinnati, Ohio.

Cairncross, R. A., Schunk, P. R., Baer, T. A., Rao, R. R. and Sackinger, P. A. 2000 A finite element method for free surface flows of incompressible fluids in three dimensions. Part I. Boundary fitted mesh motion. *Int. J. Numer. Meth. Fluids* **33** 375–403.

Chen, K. S., Schunk, P. R. and Sackinger, P. A. 1995 Finite element analyses of blade and slot coating flows using a Newton-Raphson pseudo-solid domain mapping technique and unstructured grids. Proceedings of the TAPPI Conference.

Davis, T. A. and Duff, I. S. 1997 An unsymmetric-pattern multifrontal method for sparse LU factorization. *SIAM J. Matrix Analysis and Applications* **18** 140–158.

Gartling, D. K., Hickox, C. E. and Givler, R. C. 1996 Simulations of coupled viscous and porous flow problems. *Comp. Fluid Dynamics* **7** 23–48.

Gates, I. D., Labreche, D. A. and Hopkins, M. M. 2000 Advanced capabilities in GOMA 3.0 - augmenting conditions, automatic continuation and linear stability analysis. *Technical Report SAND2000-2465* Sandia National Laboratories, Albuquerque, NM (United States).

Heroux, M. A. 1992 A proposal for a sparse BLAS toolkit. *Technical Report TR/PA/92/90* CERFACS.

Hood, P. 1976 Frontal solution program for unsymmetric matrices. *Int. J. Numer. Meth. Engr.* **10** 379–399.

Hughes, T. J. R. and Franca, L. P. 1987 A new finite element formulation for computational fluid dynamics: VII The Stokes problem with various well-posed boundary conditions: Symmetric formulations that converge for all velocity/pressure spaces. *Comput. Methods Appl. Mech. Engrg.* **59** 85–99.

Hutchinson, S. A., Shadid, J. N. and Tuminaro, R. S. 1995 Aztec Users Guide Version 1.0. *Technical Report SAND95-1559* Sandia National Laboratories, Albuquerque, NM (United States).

Kernighan, B. W. and Ritchie, D. M. 1988 *The C programming language.* 2nd edition PTR Prentice Hall, New York, NY.

Kistler, S. F. and Scriven, L. E. 1983 Coating flow. In J. M. Pearson and S. M. Richardson (eds), *Computational Analysis of Polymer Processing* Applied Science Publishers, London, UK.

Kundert, K. S. 2003 *Sparse user's guide - A sparse linear equation solver.*

Rao, R. R., Mondy, L. A., Schunk, P. R., Sackinger, P. A. and Adolf, D. B. 2001 Verification and validation of encapsulation flow models in GOMA, Version 1.1. *Technical Report SAND2001-2947* Sandia National Laboratories, Albuquerque, NM (United States).

Sackinger, P. A., Schunk, P. R. and Rao, R. R. 1996 A Newton-Raphson pseudo-solid domain mapping technique for free and moving boundary problems: A finite element implementation. *J. Comp. Phys.* **125** 83–103.

Salinger, A. G., Bou-Rabbe, N. M., Burroughs, E. A., Lehoucq, R. B., Palowski, R. P., Romero, L. A. and Wilkes, E. D. 2002 LOCA 1.0: Theory and implementation manual. *Technical Report SAND2002-0396* Sandia National Laboratories, Albuquerque, NM (United States).

Schoof, L. A. and Yarberry, V. R. 1992 EXODUS II: A finite element data model. *Technical Report SAND92-2137* Sandia National Laboratories, Albuquerque, NM (United States).

Schunk, P. R. 2000 TALE: An Arbitrary Lagrangian-Eulerian approach to fluid-structure interaction problems. *Technical Report SAND2000-0807* Sandia National Laboratories, Albuquerque, NM (United States).

Schunk, P. R., Heroux, M. A., Rao, R. R., Baer, T. A., Subia, S. R. and Sun, A. C. 2001 Iterative solvers and preconditioners for fully-coupled finite element formulations of incompressible fluid mechanics and related transport problems. *Technical Report SAND2001-3512J* Sandia National Laboratories, Albuquerque, NM (United States).

Schunk, P. R. and Rao, R. R. 1994 Finite element analysis of multicomponent two-phase flows with interphase mass and momentum transport. *Int. J. Numer. Meth. Fluids* **18** 821–842.

Schunk, P. R., Sackinger, P. A., Rao, R. R., Cairncross, R. A., Baer, T. A. and Labreche, D. A. 1997 GOMA 2.0- A full-Newton finite element program for free and moving boundary Problems with coupled fluid/solid momentum, energy, mass, and chemical species transport: Users guide. *Technical Report SAND97-2404* Sandia National Laboratories, Albuquerque, NM (United States).

Schunk, P. R. and Shadid, J. N. 1992 Iterative solvers in implicit finite element codes. *Technical Report SAND92-1158* Sandia National Laboratories, Albuquerque, NM (United States).

Shadid, J. N., Moffat, H. K., Hutchinson, S. A., Hennigan, G. L., Devine, K. D. and Salinger, A. G. 1995 MPSalsa: A finite element computer program for reacting flow problems, Part 1 - Theoretical development. *Technical Report SAND95-2752* Sandia National Laboratories, Albuquerque, NM (United States).

Tuminaro, R. S., Heroux, M. A., Hutchinson, S. A. and Shadid, J. N. 1999 Official Aztec Users Guide Version 2.1. *Technical Report SAND99-8801J* Sandia National Laboratories, Albuquerque, NM (United States).